An 8.29 mm^2 52 mW Multi-Mode LDPC Decoder Design for Mobile WiMAX System in 0.13 μm CMOS Process

Xin-Yu Shih, Cheng-Zhou Zhan, Cheng-Hung Lin, and An-Yeu (Andy) Wu

***Abstract*—This paper presents a multi-mode decoder design for Quasi-Cyclic LDPC codes for Mobile WiMAX system. This chip can be operated in 19 kinds of modes specified in Mobile WiMAX system, including block sizes of 576, . . ., 2304. There are four proposed design techniques: reordering of the base matrix, overlapped operations of main computational units, early termination strategy and multi-mode design strategy. Based on overlapped decoding mechanism, the decoding latency can be reduced to 68.75% of non-overlapped method, and the hardware utilization ratio can be enhanced from 50% to 75%. Besides, the proposed early termination strategy can dynamically adjust the number of iterations when dealing with communication channels of different SNR values. The proposed multi-mode LDPC decoder design is implemented and fabricated in TSMC 0.13 μm 1.2 V 1P8M CMOS technology. The maximum operating frequency is measured 83.3 MHz and the corresponding power dissipation is 52 mW. The core size is 4.45 mm^2 and the die area only occupies 8.29 mm^2.**

***Index Terms*—LDPC codes, low power and early termination, mobile WiMAX, multi-mode design.**

I. Introduction

LOW-DENSITY parity-check (LDPC) codes, which are one kind of linear block codes, were first introduced by Gallager in 1962 [1] and were verified to possess superior error-correcting capabilities. In 1996, LDPC codes were rediscovered and simulated by MacKay [2]. With the advanced technology, the interests in LDPC codes have been dramatically increased because their excellent error-correcting performance is much closer to the Shannon limit. Hence, LDPC codes have widely adopted by most advanced wire-line and wireless communication systems, such as IEEE 802.3an, 802.11n, and 802.16e systems. Recently, there are many interesting research works on LDPC codes and LDPC decoder architecture, including fully parallel method [3], partially parallel method [4], grouped sequentially method [5], and completely sequentially method [6]. From these state-of-arts, parallelism provides the design trade-off between the hardware cost and the system throughput. However, there still exist many challenges, such as multi-mode design, high routing complexity, large chip area, and high power consumption. As shown in Table I, this paper presents the design and implementation of the 19-mode LDPC decoder for Mobile WiMAX, IEEE 802.16e system [7], [8].

TABLE I
19 Modes for LDPC Decoder Design

Mode	Codeword n (bits)	Information bits k (bits)	Mode	Codeword n (bits)	Information bits k (bits)
1	576	288	11	1536	768
2	672	336	12	1632	816
3	768	384	13	1728	864
4	864	432	14	1824	912
5	960	480	15	1920	960
6	1056	528	16	2016	1008
7	1152	576	17	2112	1056
8	1248	624	18	2208	1104
9	1344	672	19	2304	1152
10	1440	720			

In this paper, we propose four design techniques for LDPC codes in Mobile WiMAX system: 1) reordering of the base matrix and 2) overlapped operations of computational units for increasing hardware utilization ratio; 3) early termination strategy for flexible decoding throughput and low power consumption; 4) reconfigurable architecture for multi-mode design. Besides, in order to reduce routing complexity and power consumption, we propose an efficient checkerboard layout scheme in chip implementation. Using TSMC 0.13 μm CMOS technology, the core area of this 19-mode decoder chip is only 4.45 mm^2 with die size of 8.29 mm^2. It can be measured at 83.3 MHz with only 52 mW power consumption.

The remainder of this paper is organized as follows. In Section II, we introduce low-density parity-check codes and two popular decoding algorithms, Sum-Product Algorithm (SPA) and Min-Sum Algorithm (MSA). In Sections III and IV, respectively, the proposed design techniques and VLSI architecture for 19-mode LDPC decoder design are demonstrated. The chip implementation via TSMC 0.13 μm CMOS technology is shown in Section V. Conclusions are presented in Section VI.

II. Low-Density Parity-Check (LDPC) Codes

A. Low-Density Parity-Check Codes

The (n, k) LDPC codes are one kind of block codes, where k and n represent the number of information bit and codeword,

Manuscript received February 5, 2007; revised September 13, 2007. This work was supported by the National Science Council of Taiwan under Grant NSC 94-2220-E-002-025 and the Ph.D. fellowship program of MediaTeK Education Foundation. Chip fabrication was supported by Chip Implementation Center (CIC). Part of the material in this paper has been presented at the IEEE Symposium on VLSI Circuits, Japan, June, 2007.

The authors are with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 10617, R.O.C. (e-mail: genius@access.ee.ntu.edu.tw; andywu@cc.ee.ntu.ac.tw).

Digital Object Identifier 10.1109/JSSC.2008.916606

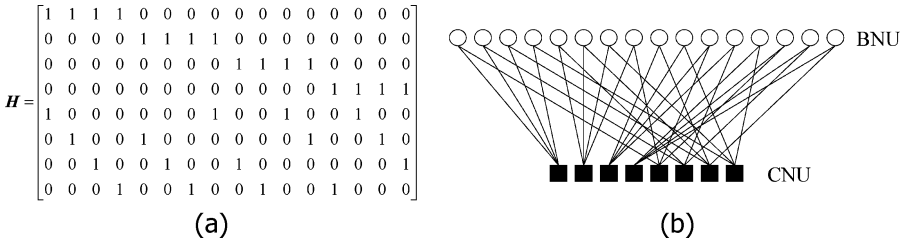


Fig. 1. (16, 8) LDPC codes (a) parity check matrix $\boldsymbol{H}_{8*16}$, and (b) Bipartite graph.

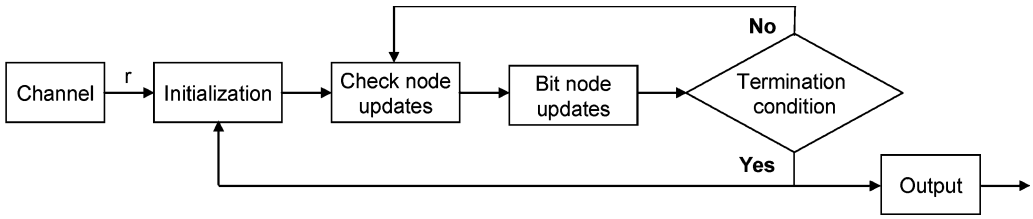


Fig. 2. Decoding flowchart of belief propagation decoding algorithm.

respectively. LDPC codes are defined by a very sparse parity check matrix $\boldsymbol{H}$, which the most elements are 0's and only several elements are 1's. The parity check matrix $\boldsymbol{H}_{(n-k)*n}$ of (n, k) LDPC codes has $(n - k)$ rows and n columns and the corresponding code rate R is defined as k/n. Fig. 1(a) shows that the parity check matrix $\boldsymbol{H}_{8*16}$ of (16,8) LDPC codes with code rate of 1/2 has 8 rows and 16 columns.

In the decoding aspect, a parity check matrix can be mapped into a bipartite graph [9] or a factor graph [10], as illustrated in Fig. 1(b). Rows and columns of the parity check matrix $\boldsymbol{H}$ can be mapped to check node units (CNUs) and bit node units (BNUs), respectively. Based on the location of the elements of '1' in the parity check matrix $\boldsymbol{H}$, it is very straightforward to make the straight connection between corresponding CNU and BNU. The direct connection can represent the mutual information transmission among CNUs and BNUs.

For code construction, there are various ways to construct the parity check matrix $\boldsymbol{H}$. Although the decoding performance of any structured LDPC codes [11] is worse than that of random ones [12], the benefit of structured method is to make the designers much easier to handle the location of 1's in the parity check matrix $\boldsymbol{H}$. In addition, the regularity in structure-based method can provide more help on the decoder design. Among many structured LDPC code construction methods, Quasi-Cyclic LDPC codes [13] become the most popular in the modern advanced communication systems, such as Mobile WiMAX system.

B. Decoding Algorithm

It is very popular to use belief propagation decoding algorithm [14] for the LDPC decoder. The decoding flowchart of belief propagation decoding algorithm can be illustrated in Fig. 2. First, the message is initialized as the received data r from communication channel. Second, the check nodes perform updating by collecting the message from connected bit nodes. As shown in Fig. 3(a), the updated message from the check node C to the bit node B_2 is related to the message from the bit nodes B_1, B_3, and B_4. Similarly, the bit nodes perform updating by collecting the message from connected check nodes. As shown in Fig. 3(b), the updated message from the check node B to the bit node C_2 is related to the message from the bit nodes C_1, C_3, and C_4. After one round of check node and bit node updates, the termination condition should be checked. If the termination criterion is satisfied, the block of codeword is decoded completely. Otherwise, the decoding operations would be performed iteratively until the termination criterion is met.

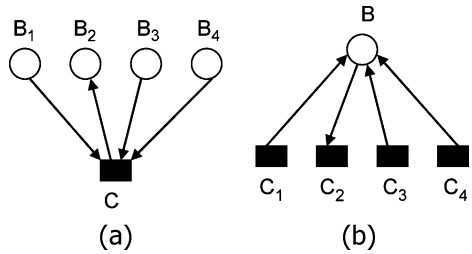


Fig. 3. (a) Check node updates, and (b) bit node updates.

Generally specking, there are two prevalent decoding algorithms for LDPC codes, including *Sum-Product Algorithm (SPA)* and *Min-Sum Algorithm (MSA)* [14]. The benefits of the former are high-precision message passing and excellent error-correcting performance. Although it possesses the outstanding correcting ability, it is not suitable for the VLSI design due to more complex mathematical computation and higher hardware complexity. However, the latter has lower computation complexity with little performance scarification. Thus, the MSA is a better candidate for hardware implementation.

III. Proposed Design Techniques for LDPC Decoder Design

In this section, we propose four design techniques, including reordering of the base matrix, overlapped operations of BNUs and CNUs, early termination strategy, and multi-mode design strategy.

A. Reordering of the Base Matrix

Fig. 4 shows the base parity check matrix $\boldsymbol{H}$ with code rate of 1/2 defined in Mobile WiMAX system. The base matrix $\boldsymbol{H}$ with

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1		94	73						55	83			7	0										
2		27				22	79	9				12		0	0									
3				24	22	81		33				0			0	0								
4	61		47						65	25						0	0							
5			39				84			41	72						0	0						
6					46	40		82				79	0					0	0					
7			95	53						14	18								0	0				
8		11	73				2			47										0	0			
9	12				83	24		43				51									0	0		
10						94		59			70	72										0	0	
11			7	65					39	49													0	0
12	43					66		41				26	7											0

Fig. 4. Base parity check matrix $\boldsymbol{H}$ with code rate of 1/2 in Mobile WiMAX system [7].

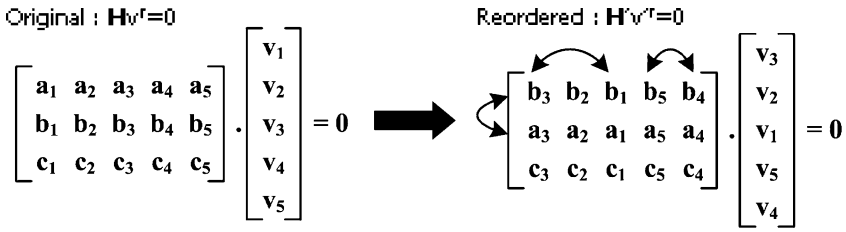


Fig. 5. Relationship of parity check matrix and codeword.

elements of blank, zero and non-zero integer has 12 rows and 24 columns. The element of blank can be expanded as a $p * p$ zero matrix. The element of zero can be expanded as a $p * p$ identity matrix. The element of non-zero integer, $s(i,j)$, can be expanded as a $p * p$ cyclic-shifted matrix with shifted value of $s(p,i,j)$, which can be calculated by

$$s(p,i,j) = \left\lfloor \frac{s(i,j) * p}{96} \right\rfloor \tag{1}$$

where the expanding factor p can be 24, 28, ..., 92 and 96. If p is chosen as 96, $s(p,i,j)$ is the same as $s(i,j)$.

By inspecting (1), the elements in the lower-left or upper-right parts are not all blank, i.e., zero matrixes. The bit node operation of first column needs to wait until the completion of the check node operations. On the other hand, the check node operation of first row needs to wait until the completion of the bit node operations. In order to reduce the idle time among check node and bit node operation, it is easy to improve the situation by reordering the rows and columns of $\boldsymbol{H}$ [15]. Reordering the rows and columns does not make any influence on the error-correcting performance. As shown in Fig. 5, the row reordering makes no change on codeword and the column reordering only makes the bit-level re-permutation in one codeword. In other words, the relationship of original codeword v and reordered codeword v' can be directly obtained according to the column reordering. There are four advantages in this reordering technique: (1) convenience of overlapped check node and bit node updates; (2) preservation of data locality; (3) no penalty for the extra hardware cost or executed clock cycles; (4) no decoding performance degradation.

In [15], the reordering steps are performed on the whole parity check matrix $\boldsymbol{H}$. The benefits are that the idle time can be reduced to minimum and the overlapped period can be maximized. Unfortunately, the reordered parity check matrix would not keep the quasi-cyclic characteristics and the code is not Quasi-Cyclic LDPC codes any more. The location of 1's would lose the regularity and the reordered parity check matrix is close to the random construction. Furthermore, since the non-zero sub-matrixes in reordered parity check matrix are not cyclic-shifted matrixes, it is impossible to directly map each non-zero sub-matrix into the individual memory bank in hardware architecture. The difficulty of corresponding hardware design would become more complicated. That is, this method influences not only the code property but also hardware complexity.

Instead, we only reorder the base matrix by the algorithm in [15] and new reordered parity check matrix $\boldsymbol{H_r}$ would be shown in Fig. 6. The relationship of new and original row index can be obtained as the most left side in Fig. 6. And the relationship of new and original column index can be obtained as the most upper side in Fig. 6. In this way, not only the elements in the lower-left and upper-right parts of $\boldsymbol{H_r}$ are all zero matrixes, but also the characteristic of Quasi-Cyclic LDPC codes for Mobile WiMAX system can be maintained. In hardware implementation, the reordering algorithm only changes the control signals in the control unit without any penalty for extra hardware cost or executed clock cycles. The modification for collecting the channel values in the beginning is to rearrange the index of input buffers and message-passing memory banks on the timing diagram. In the following decoding state, the memory access is also arranged by the modified control signals.

B. Overlapped Operations of BNUs and CNUs

In order to reduce the hardware cost, we can use the partially-parallel method [16] to realize the LDPC decoder. But the scarifications are the reduction of throughput and increase of decoding latency. Take (2304, 1152) LDPC codes for example. If we choose the folding factor as $96 * 3$ in Table II, it not only achieves the minimum throughput requirement for Mobile WiMAX system, but

old →		17	20	14	9	7	2	3	10	18	21	16	4	13	1	6	11	22	23	19	15	24	5	8	12
↓	new	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	1			0	55		94	73	83					7											
4	2	0			65			47	25			0			61										
5	3	0				84		39	41	0							72								
8	4		0			2	11	73	47		0														
7	5		0					95	14				53				18			0					
11	6				39			7	49				65						0			0			
2	7			0		79	27									22					0			9	12
3	8											0	24			81					0		22	33	0
10	9															94	70	0	0					59	72
12	10													7	43	66						0		41	26
6	11									0				0		40				0			46	82	79
9	12										0				12	24		0					83	43	51

Fig. 6. Reordered parity check matrix $\boldsymbol{H}_r$ with folding factor of 288.

TABLE II
COMPARISON OF HARDWARE UTILIZATION AND DECODING LATENCY

Folding factor		96	96*2	96*3	96*4	96*6
HW	Number of BNUs	24	12	8	6	4
	Number of CNUs	12	6	4	3	2
Hardware utilization	Non-overlapped	50%	50%	50%	50%	50%
	Overlapped	-	-	75%	66.67%	75%
Executed clock cycles	Non-overlapped	96*8	96*4*8	96*6*8	96*8*8	96*12*8
	Overlapped	-	-	96*(4*8+1)	96*(6*8+1)	96*(8*8+2)
	Reduce ratio	-	-	31.25%	23.44%	31.25%

also performs overlapped operations of BNUs and CNUs [17], [18] by previous reordering scheme. Hence, there are three benefits in this technique, including idle time reduction of CNUs and BNUs, enhancement of hardware utilization ratio, and reduction of decoding latency.

In Fig. 6, the bold horizontal and vertical lines illustrate the folding boundary for row and column operations, respectively. Since the folding factor is chosen as 288, there are only four CNUs and eight BNUs in hardware implementation. For simplicity, we assume that one block row and one block column consist of 96 rows and 96 columns, respectively. In the check node operations, four rows can be concurrently performed in one clock cycle. It needs 96 clock cycles to complete the check node operations of 4 block rows. Totally, it needs 288 clock cycles to complete all the row operations. On the other hand, for the bit node operations, eight columns can be concurrently performed in one clock cycle. It also needs 96 clock cycles to complete the bit node operations of 8 block columns. Similarly, it also needs 288 clock cycles to complete all the column operations.

In Fig. 7(a), the BNUs and CNUs work alternatively and the property of non-overlapped operations would result in lower hardware utilization ratio equaling to 50% and longer decoding latency. Since lower-left parts in Fig. 6 are all zero matrixes, the bit node operations of the first 8 block columns can be early performed after the check node operations of the first 8 block rows. The reason is that the needed exchanged information has been updated without affected by the check node operations of the last 4 block rows. On the other hard, the check node operations of the first 4 block rows can be early performed after the bit node operations of the first 16 block columns since upper-right parts are all zero matrixes. Similarly, the reason is that the needed exchanged information is ready without affected by the bit node operations of the last 8 block columns. By applying this technique, the decoding latency can be reduced as illustrated in Fig. 7(b).

In mathematical analysis, we assume that the predefined maximum number of the iterations is $Iter$. Non-overlapped and overlapped method need $[96*6*Iter]$ and $[96*(4*Iter+1)]$ clock cycles, respectively. The reduced ratio of executed clock cycles increases as the number of iterations increases. If $Iter$ equals to 8, the reduced ratio is 31.25% and the decoding latency shrinks to 68.75% of non-overlapped method. Besides, the hardware of CNUs or BNUs is in the idle mode of one-fourth decoding latency in average. In the proposed overlapped method, the idle time can be reduced by 66% and the hardware utilization ratio can be enhanced from 50% to 75%.

C. Early Termination Strategy

Traditionally, there are two common methods to determine the decoding terminations of one pattern of codeword. The two inefficient methods without dynamic adjustment would lead to longer decoding latency, lower decoding throughput, and larger power consumption. One is to check the valid codeword v with the equations $\boldsymbol{H}v^T = 0$. This method would take much hardware cost for the checked equations depended on the number of rows in $\boldsymbol{H}$. The other is to only set maximum number of decoding iterations. Although the method is very simple and quite easy to implement, it is very inefficient and inflexible to take the same decoding iterations for different communication environment, especially for the high SNR channels or less-noise wire-line environment.

In addition to the conventional methods, there are lots of research works in the literature about early termination recently

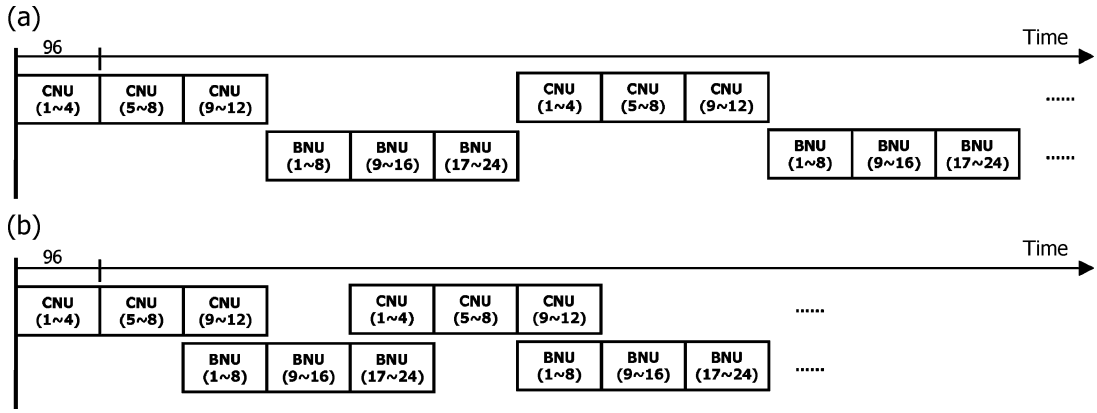


Fig. 7. Timing diagram: (a) non-overlapped method, and (b) overlapped method.

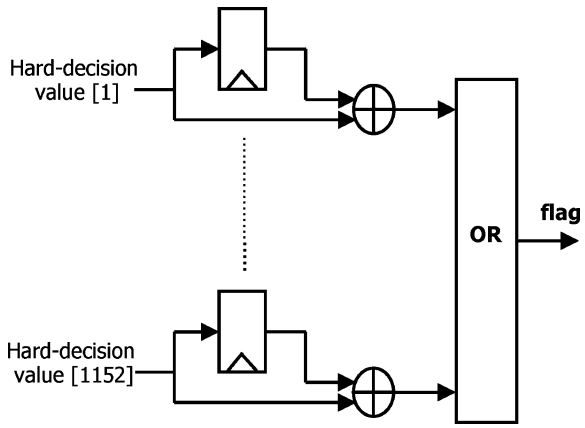


Fig. 8. Hardware design of the proposed early termination strategy (ET).

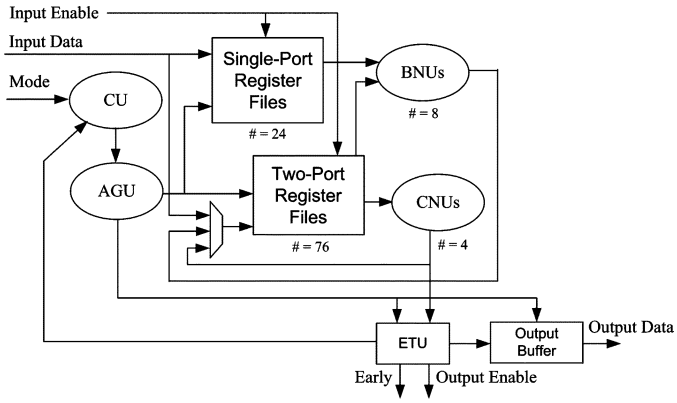


Fig. 9. Block diagram of multi-mode LDPC decoder design.

[19], [20]. In [19], the decoding blocks are divided into two groups, such as decodable and undecodable blocks. For decodable blocks, the first common method as previously mentioned can be performed. For undecodable blocks, the stopping criteria are relative to the variable node reliability and the decoding threshold. Moreover, in [20], the early termination criteria are based on the convergence of the mean magnitude (CMM) with two parameters, magnitude threshold and depth factor. Although these two methods have no decoding performance degradation, they need the mass mathematical operations and statistical properties of all the decoded messages. Unfortunately, it is not suitable for the hardware realization and chip implementation. Instead, we propose a more robust and flexible early termination strategy in accordance with the hard-decision-aided (HDA) stopping criteria [21] for Turbo decoding. The early termination strategy can not only dynamically adjust the number of iterations when dealing with communication channels of different SNR, but also be very adequate for the low-cost and low-power hardware implementation.

The early termination strategy is to store the hard-decision values by observing the sign values of *Log-Likelihood Ratio* (LLR) and compare the decoded results in two successive iterations. If all the decoded bits of this iteration are the same as those of previous iteration, the decoder is informed to terminate operations. Otherwise, the mechanism would check whether the predefined maximum number of iterations is met or not. By the way, it does not take extra executed clock cycles to apply the checked mechanism. Since LDPC codes defined in Mobile WiMAX system are systematic codes, which the information bits directly exist in the codeword, it is more efficient to only check the information bits. For (2304, 1152) LDPC codes, we only need to check the 1152 information bits instead of the total 2304 bits.

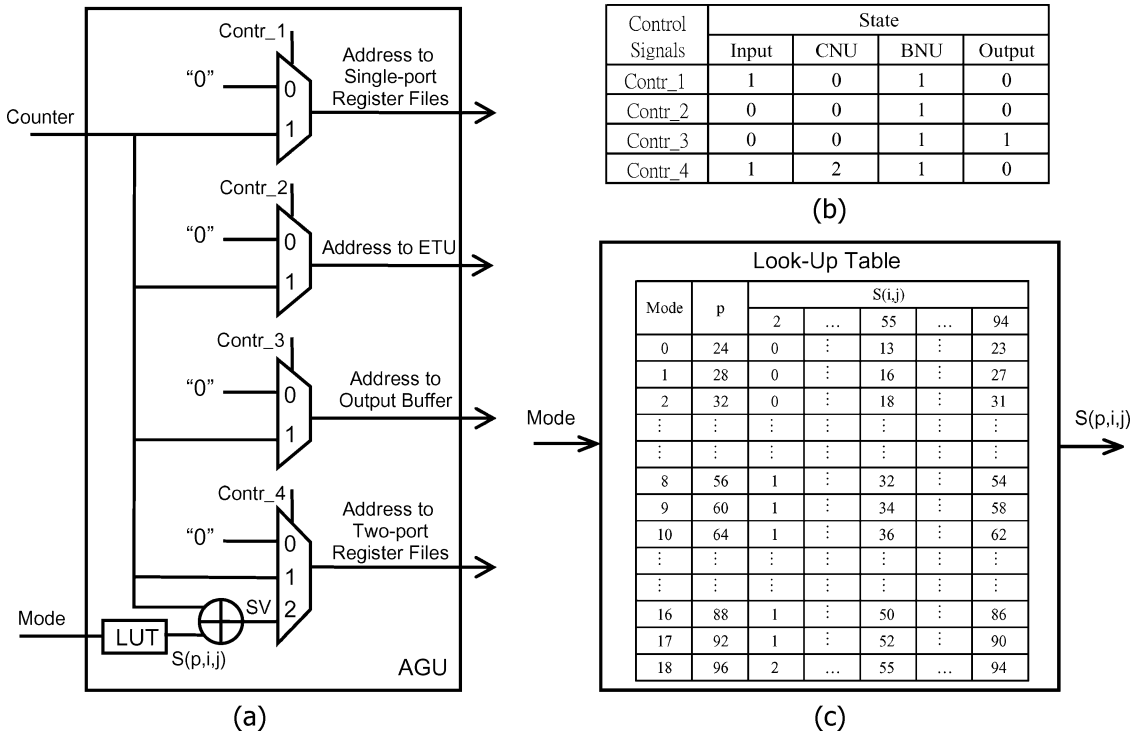
In hardware implementation, each checked bit is easily implemented by one D flip-flop and one XOR logic-gate as illustrated in Fig. 8. The *flag* signal is used to check the equality of decoded bits in the two successive iterations. If *flag* is true, the decoded results in two successive iterations are different. Otherwise, the decoded bits in two successive iterations are identical. Therefore, the termination criterion can be specified as follows,

1) *Flag* is false.
2) *Flag* is true and the predefined maximum number of iterations is met.

Once the terminated condition is met, the decoder would output the hard-decision values in original sequence. Thus, one pattern of codeword is decoded completely.

D. Multi-Mode Design Strategy

In order to achieve the multi-mode design, we propose the reconfigure architecture as shown in Fig. 9. Since we find out the relationship of 19 parity check matrices in Mobile WiMAX system, the multi-mode design can be performed



Control Signals	State			
	Input	CNU	BNU	Output
Contr_1	1	0	1	0
Contr_2	0	0	1	0
Contr_3	0	0	1	1
Contr_4	1	2	1	0

(b)

Look-Up Table

Mode	p	S(i,j)				
		2	…	55	…	94
0	24	0	⋮	13	⋮	23
1	28	0	⋮	16	⋮	27
2	32	0	⋮	18	⋮	31
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	56	1	⋮	32	⋮	54
9	60	1	⋮	34	⋮	58
10	64	1	⋮	36	⋮	62
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
16	88	1	⋮	50	⋮	86
17	92	1	⋮	52	⋮	90
18	96	2	…	55	…	94

(c)

Fig. 10. AGU design: (a) architecture, (b) control signals, and (c) look-up table.

by *Address Generation Unit* (AGU) without modifying any storage elements and computational units. According to the counter signals from *Counter Unit* (CU), it is easy for AGU to generate the necessary address for Single/Two-port Register Files, *Output Buffer*, and *Early Termination Unit* (ETU). As the hardware architecture demonstrated in Fig. 10(a), the address for the Two-port Register Files have three choices, such as '0', *Counter*, and SV. In the different states, the address is determined by the control signal as shown in Fig. 10(b). Especially in the CNU state, the candidate address is generated by adding *Counter* and $s(p,i,j)$. The corresponding value of $s(p,i,j)$ would be decided by *Mode* and $s(i,j)$ as the Look-up Table in Fig. 10(c). Based on (1), we can easily built the Look-up Table for $s(p,i,j)$. Instead of directly implementing the divider circuit, the Look-up Table can save more combinational circuit area for the low-cost design. Besides the address for the Two-port Register Files, the address for the other modules is simply determined as '0' or counter by the corresponding control signals.

Fig. 11(a), (b) shows the memory access and counter range for different block sizes of LDPC codes. For (576, 288) LDPC codes, the number of used memory entries is only 24, which is one-fourth of memory size. The corresponding counter in CU counts from 0 to 23. Similarly, the ratio of used memory entries and the counter range increase as block size increases. Thus, as regards (2304, 1152) LDPC codes, all of the entries in the memory banks are fully used and the counter circularly counts from 0 to 95.

IV. Proposed Decoder Architecture

In order to determine wordlength for mutual information, we perform six simulation cases for different block sizes with different SNR. The block sizes are chosen as 576, 1440, and 2304 and the SNR are 3.5 and 4.5 dB. The determination procedure can be divided into two steps.

Step 1) First, we choose the fractional part as 4 bits. The integer part would be chosen as $1 \sim 6$ bits. As shown in Fig. 12(a), the bit error rate (BER) is almost saturated when the overall word-length is 8-bit, including 4-bit integer part and 4-bit fractional part. Second, we adjust the bit number of the fractional part. The simulation results also have the similar saturation phenomenon when the integer part is chosen as 4 bits. In other words, the BER would be saturated with the increase of the integer part no matter what the bit number of the fractional part is. The only difference is that the saturated BER would be higher as the bit number of the fractional part decreases.

Step 2) After the integer part is determined as 4 bits, the fractional part would be chosen as $2 \sim 7$ bits. In Fig. 12(b), BER is almost saturated when the overall word-length is 8-bit, including 4-bit fractional part and 4-bit integer part. Therefore, we choose 8-bit fixed-point representation as wordlength for the exchanged information between CNUs and BNUs in hardware implementation.

After performing the fixed-point analysis, we can construct the needed computation units, such as CNUs and BNUs, with single-port and two-port register files.

A. Architecture for CNU Design

To perform the check node updating equations in Min-Sum Algorithm, we should first search for the minimum and second minimum among the received data. As demonstrated in Fig. 13(a), the common part design of 2-input CNU, $\mathrm{CNU\_common\_2}$, is implemented with one adder and one multiplexer. If $data2$ is larger than $data1$, min and $min2$ are $data1$ and $data2$, respectively. Otherwise, we can just exchange the two received data to min and $min2$. For common

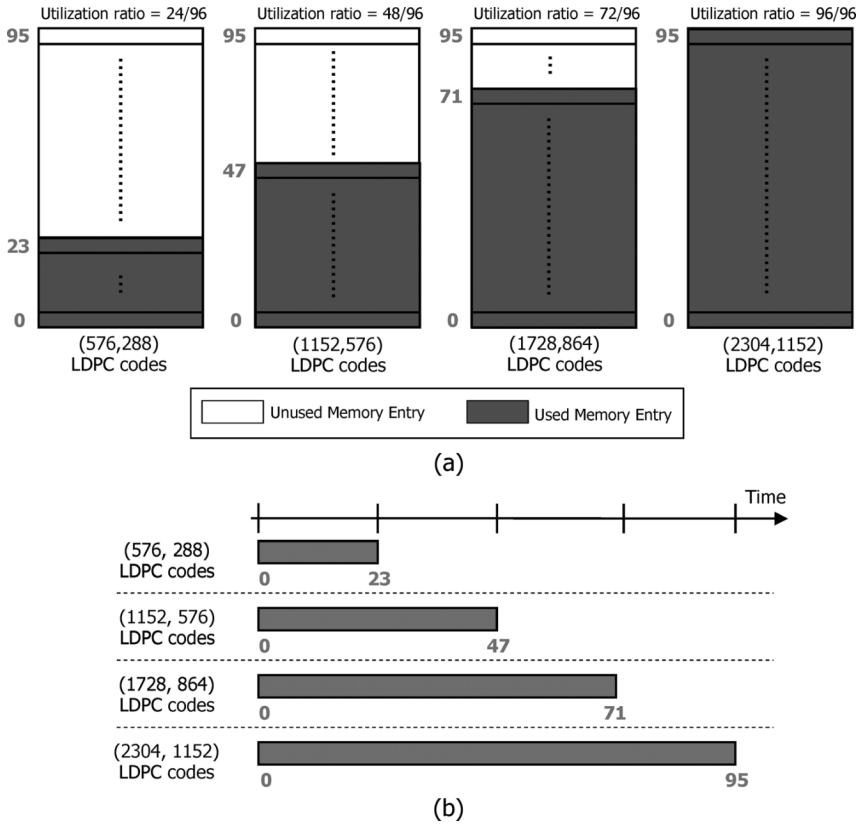


Fig. 11. Multi-mode design: (a) memory access, and (b) counter range.

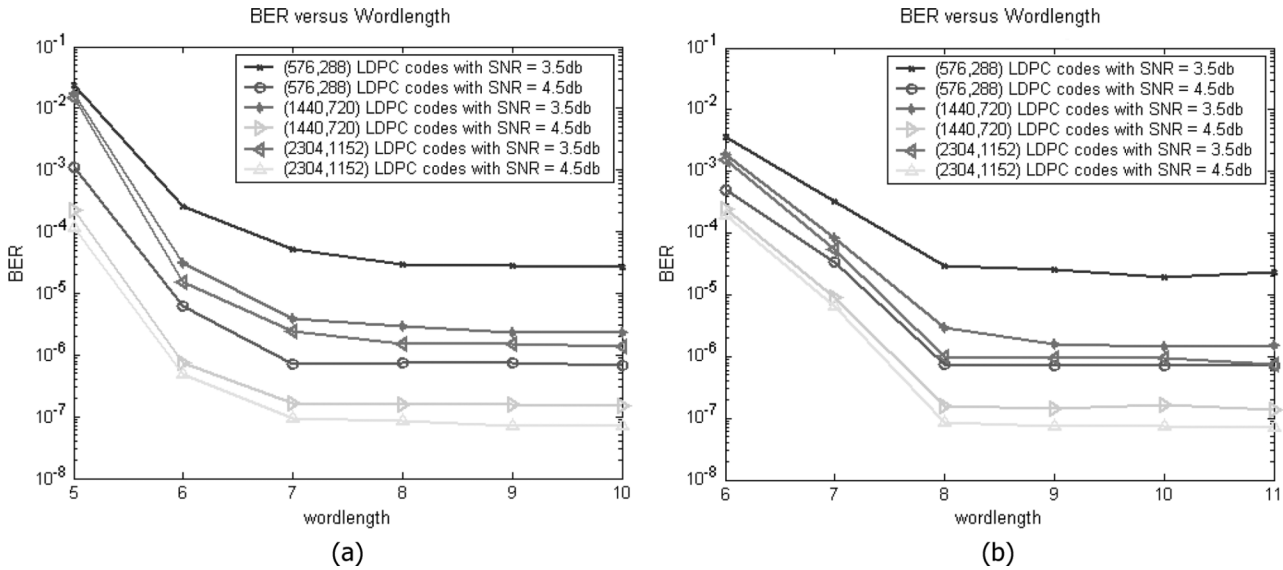


Fig. 12. BER versus word-length: (a) the fractional part is 4-bit, and (b) the integer part is 4-bit.

part design of 3-input CNU, CNU_common_3, the design can be implemented with three adders, five multiplexers and several simple logical gates, as shown in Fig. 13(b).

Based on the construction of CNU_common_2 and CNU_common_3, the design which compares more than three input data can be easily realized in hierarchal methods. As shown in Fig. 13(c), the common part design of 6-input CNU, CNU_common_6, is realized with two blocks of CNU_common_3 in the first stage and three blocks of CNU_common_2 in the following two stages. Similarly, the common part design of 7-input CNU, CNU_common_7, is realized with four blocks of CNU_common_3 as shown in Fig. 13(d). Once the common part designs are completed, we can make good use of them and build needed architecture in the CNU design. Owing to the analysis demonstrated in Section III-B, we only need four CNUs, including two 6-input and two 7-input processing elements. Owing to the similar design methodology, we only show the architecture of the 7-input CNU, as illustrated in Fig. 14.

In the overall hardware implementation, both two kinds of CNUs need the calculation of the absolute values for all input data by the block of $abs()$ in the beginning. The second step is to pass all absolute values to common part design. Then, the block of *compare* and *select* is to compare min and $min2$ with the

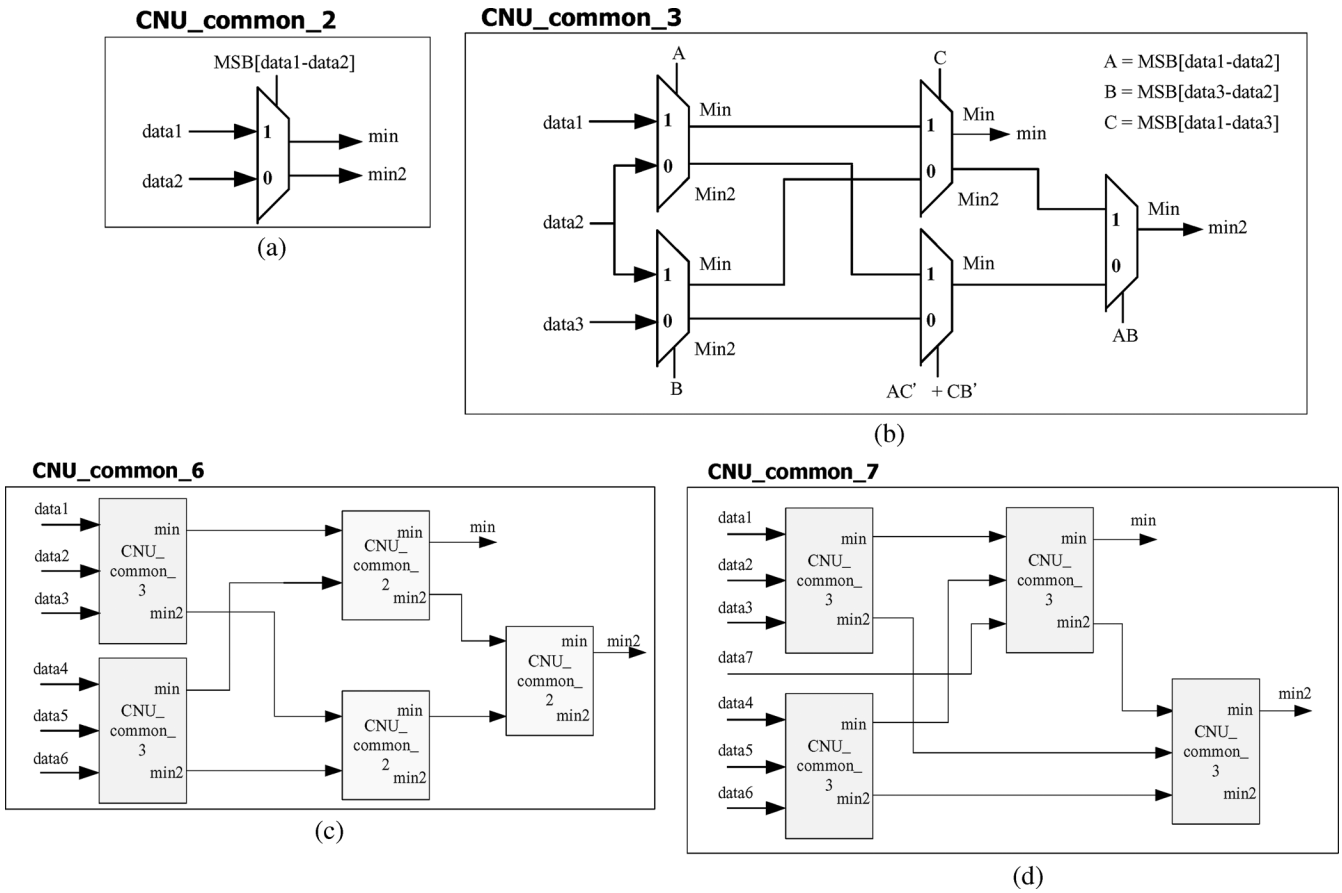


Fig. 13. Common part design: (a) 2-input, (b) 3-input, (c) 6-input, and (d) 7-input CNU.

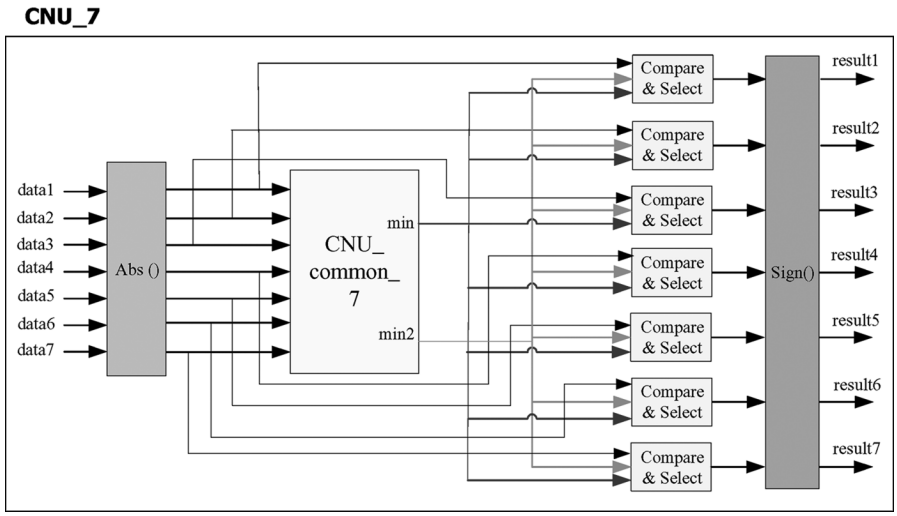


Fig. 14. Architecture of 7-input CNU.

according absolute values and choose the wanted values, min or $min2$. Finally, the block of $sign()$ is to adjust the sign of the target values and generate the accurate results.

B. Architecture for BNU Design

For the bit node updating equations in Min-Sum Algorithm, we should first calculate the summation of the received data, including the mutual information and the channel values. In Fig. 15(a), (b), the common part designs of 3-input and 4-input BNUs, BNU_common_3 and BNU_common_4, are implemented with one 3-input and three 2-input adders, respectively. Similarly, BNU_common_7 is implemented with two 2-input and two 3-input adders, as shown in Fig. 15(c). Once the common part designs are constructed, we can take them as the kernel and build needed architecture in the BNU design. Owing to the analysis demonstrated in Section III-B, we only

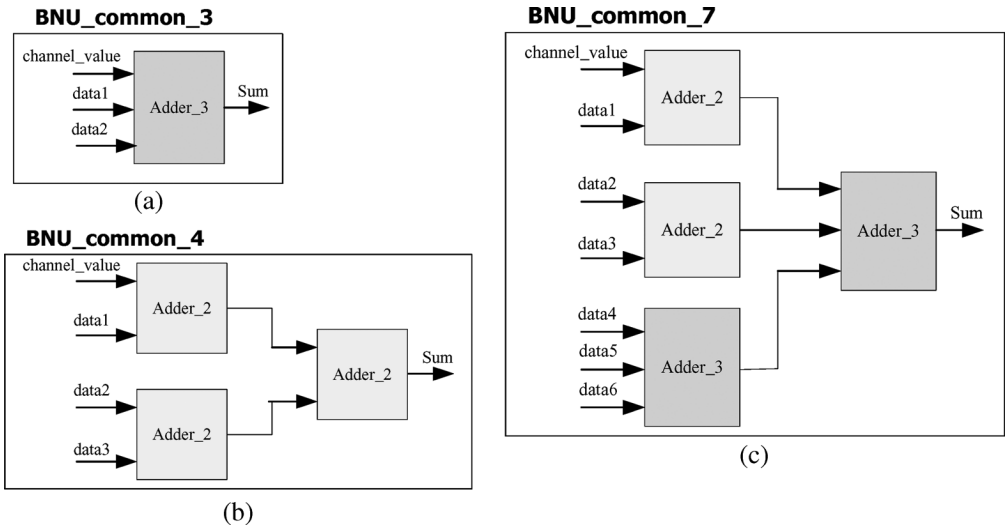


Fig. 15. Common part design (a) 3-input, (b) 4-input, and (c) 7-input BNU.

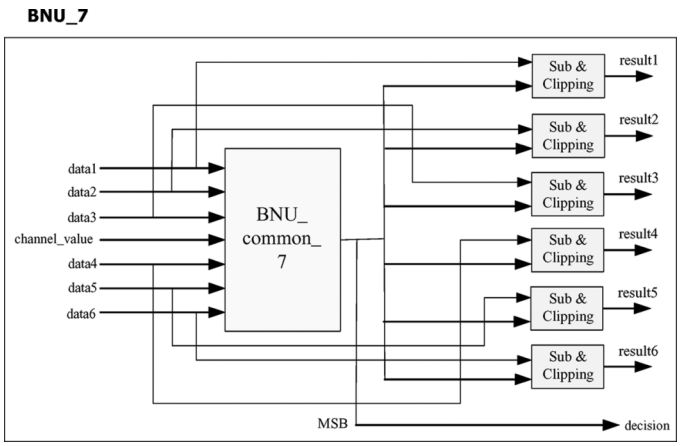


Fig. 16. Architecture of 7-input BNU.

need eight BNUs, including three 3-input, three 4-input and two 7-input processing elements. Owing to the similar design methodology, we only show the architecture of the 7-input BNU, as demonstrated in Fig. 16.

In the overall hardware implementation, all of three kinds of BNUs need the calculation of the summation of all input data first. Then, the block of *sub and clipping* is to subtract according values from the summation, perform clipping and generate the accurate results. In addition, the MSB of the summation is taken as the hard-decision value for the decoded result.

C. Block Diagram for LDPC Decoder

In order to enhance the data access parallelism, the storage elements can be implemented by 100 memory banks, which consist of 24 single-port and 76 two-port register files, instead of a central main memory. The input buffers, which store channel values for BNUs, are implemented by 24 single-port register files. The message-passing memory banks, which store exchanged information among CNUs and BNUs, are implemented by 76 two-port register files. Each of the memory banks has 96 entries, which one entry consists of 8-bit data. By activating the necessary memory banks, the operating frequency for system specification could be lower with respect to central main memory. Moreover, based on the divided memory scheme, the power consumption is reduced for the mobile application.

Fig. 9 also shows the block diagram of the LDPC decoder design. The 8-bit input data, which consists of 4-bit integer part and 4-bit fractional part, is sequentially stored in the corresponding input buffers and message-passing memory banks when the '*Input enable*' signal is asserted. After collecting all the channel values within one codeword, the overlapped operations of BNUs and CNUs are performed. Due to the folding property of BNUs, the unused input of BNU is fed as zero value to produce the correct calculated data when the number of input data is less than the hardware input ports at certain moment. The same situation would occur in CNUs but the fed value is different from that in BNUs. The unused input of CNU is fed as the maximum value in fixed-point representation to produce the correct calculated data when the number of input data is less than the hardware input ports at certain moment.

In the decoding state, the '*Early*' signal is asserted to represent the actual number of decoding iterations is less than predefined number when termination criterion is met. Therefore, the system can be early terminated with less decoding iterations and the decoding latency is reduced. However, the terminated condition is satisfied and the decoded information bits can be available. The decoded information bits, hardware-decision values of LLR, are stored in the output buffers, which are directly implemented by $12 * 96$ D flip-flops. In the output state, the '*Output enable*' signal is asserted and the information bits are grouped into 24-bit output data. After previous described procedure, the decoding of the block of pattern is accomplished.

V. Experimental Results and Chip Implementation

For Mobile WiMAX system, we have implemented multi-mode LDPC decoder design, which can perform 19 kinds of operating modes, including block sizes of 576, ..., 2304, as shown in Table I. The chip design is fabricated in

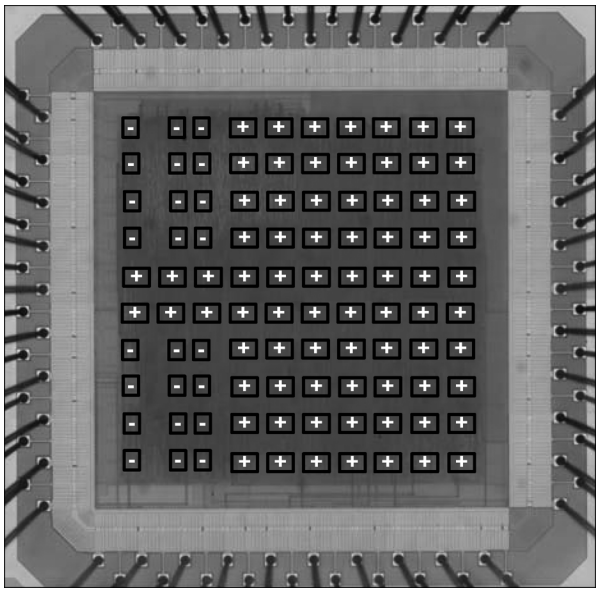


Fig. 17. Die photo of multi-mode LDPC decoder design. (− represents single-port register files and + represents two-port register files).

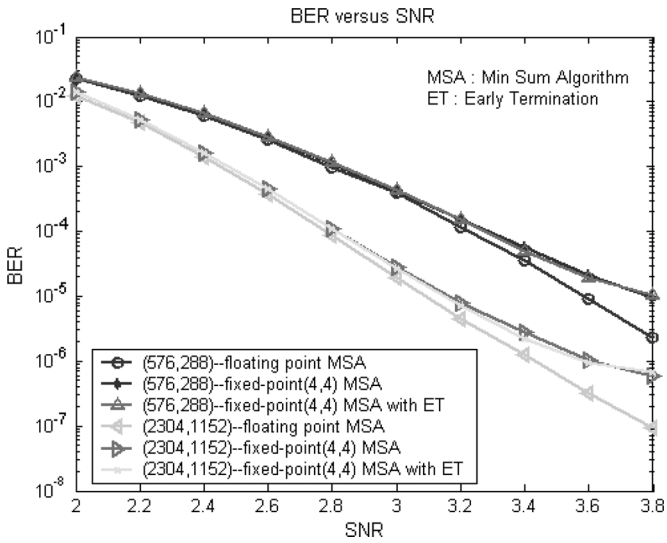


Fig. 18. Performance comparison.

TSMC 0.13 μm 1.2 V 1P8M CMOS technology and measured with Tektronix pattern generator TLA715 and logic analyzer TLA 5203. For lower routing complexity, we propose an efficient checkerboard layout scheme to arrange 100 memory banks in 10 × 10 2-D arrays as shown in Fig. 17. Compared with other types of arrangement, the method of the uniformly distributed memory banks is suitable for LDPC chip implementation. The benefits are chip area reduction and low power dissipation. Moreover, Fig. 18 shows the performance of the proposed decoding strategy with ET. Take the smallest and largest block sizes of LDPC codes for examples. The measurement results of (576, 288) and (2304, 1152) LDPC codes are exactly equivalent to the software simulation results of the fixed-point (4, 4) MSA without ET case.

The chip feature is summarized in Table III. The core size is 2.11 mm × 2.11 mm. The number of total gate counts is 420 K

TABLE III
CHIP SUMMARY

Irregular LDPC codes with Code Rate 1/2 (19 Kinds of operating modes)	
Cell Library	TSMC 0.13um 1P8M
Work Voltage	1.2V / 3.3 V
Gate Count	420 K
Core Size	2.11mm x 2.11mm (4.45 mm^2)
Die Size	2.88mm x 2.88mm (8.29 mm^2)
Operating Frequency	83.3 MHz (max)
Power Consumption	52mW @ 83.3MHz

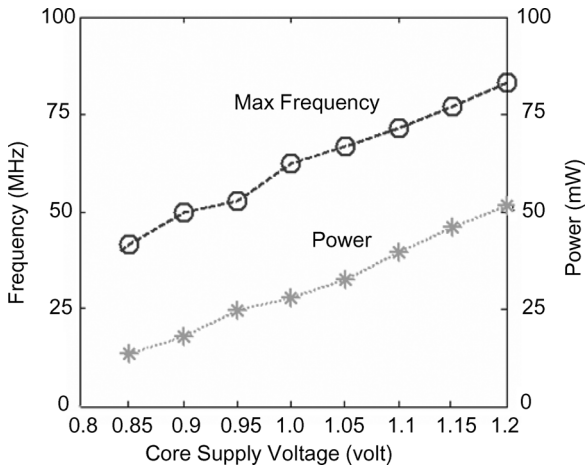


Fig. 19. Measured frequency and power of the chip design. (Under minimum sustainable supply voltage (0.85 V), the decoder chip can deliver 30 Mb/s as required in Mobile WiMAX system.)

and the die size is only 2.88 mm × 2.88 mm giving a total area of 8.29 mm^2. The maximum operating frequency is measured 83.3 MHz and the total power consumption is measured 52 mW for 19-mode LDPC decoder design. In the decoder design with early termination, the actual number of decoding iterations is at least two no matter how higher SNR is due to equality-checking of hard-decision values in the two successive iterations. As a result, the flexible throughput can be varied from 60.6 Mb/s (@ 8 iterations) to 222.2 Mb/s (@2 iterations) by dynamic adjustment. The throughput for different iteration numbers is higher than the specification of Mobile WiMAX system (30 Mb/s). Therefore, the decoding strategy provides the flexible and efficient mechanism.

In order to simultaneously consider reduction of power consumption and minimum throughput requirement for Mobile WiMAX system, we can reduce the core supply voltage to 0.85 V as shown in Fig. 19. The corresponding maximum operating frequency and power consumption are 41.67 MHz and 16 mW, respectively. In summary, the multi-mode decoder design is compared with other chip designs as shown in Table IV. Among the five chip designs, the code construction, block size specification, and process technology are very different.

TABLE IV
COMPARISON TABLE

	JSSC'02 [22]	ISCAS'05 [23]	JSSC'06 [24]	TCAS'06 [25]	This work
Multi-mode	No	No	7 modes	No	19 modes
Spec	(1024,512)	(2048,1732)	(2048,128*k) k = 8~14	(1024,512)	(96*k, 48*k) k = 6~24
Code Construction	Random	RS-based	Turbo-Interleaved	QC-based	QC-based
Technology	0.16um	0.18um	0.18um	0.18um	0.13um
Parallelism	Fully	Fully	Partial	Partial	Partial
Iterations	64	32	16	8	2 ~ 8
Area (mm^2)	52.5	17.64	14.3	10.08	8.29
Frequency	64MHz	100MHz	125MHz	200MHz	83.3MHz
Throughput	1Gbps	3.2Gbps	640Mbps	985Mbps	60~222Mbps
Power	690mW	N/A	787mW	N/A	52mW

Obviously, the chip size of the multi-mode decoder design is smaller than other arts. Up to now, the decoder design with smaller die size is the most competing design compared with existing LDPC designs presented in the literature. In addition, by applying proposed design techniques and efficient layout scheme, the power consumption is also much smaller than other works. Most important of all, the outperformance of low-cost design and low power dissipation is the urgent and necessary requirement for the Mobile WiMAX system.

VI. CONCLUSION

We propose an efficient and effective IC design strategy, which not only for Mobile WiMAX system but also for any Quasi-Cyclic LDPC codes. The 19-mode decoder design using TSMC 0.13 μm technology is measured at 83.3 MHz with power consumption of 52 mW. In summary, the LDPC decoder design features smaller chip area, higher hardware utilization, lower decoding latency, flexible decoding throughput, and lower power consumption.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 7, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 399–431, Jan. 1999.

[3] C. J. Howland and A. J. Blanksby, "Parallel decoding architectures for low density parity check codes," in *Proc. IEEE ISCAS*, May 2001, vol. 4, pp. 742–745.

[4] Z. Cui and Z. Wang, "Area-efficient parallel decoder architecture for high rate QC-LDPC codes," in *Proc. IEEE ISCAS*, May 2006, pp. 5107–5110.

[5] M. Cocco, J. Dielissen, M. Heijligers, A. Hekstra, and J. Huisken, "A scalable architecture for LDPC decoding," in *Proc. IEEE Conf. Design Automation and Test in Europe (DATE)*, Feb. 2004, vol. 3, pp. 88–93.

[6] E. Yeo, B. Nikolic, and V. Anantharam, "Architectures and implementations of low-density parity-check decoding algorithms," in *Proc. Midwest Symp. Circuits and Systems*, Aug. 2002, vol. 3, pp. 437–440.

[7] IEEE 802.16e. [Online]. Available: http://www.ieee802.org/16/tge/

[8] WiMAX Forum. [Online]. Available: http://www.wimaxforum.org/home

[9] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 399–431, Sep. 1981.

[10] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[11] J. M. F. Moura, J. Lu, and H. Zhang, "Structured low-density parity-check codes," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 42–55, Jan. 2004.

[12] E. Yeo, B. Nikolic, and V. Anantharam, "Iterative decoder architectures," *IEEE Commun. Mag.*, vol. 41, pp. 132–140, Aug. 2003.

[13] H. Zhong and T. Zhang, "Design of VLSI implementation-oriented LDPC codes," in *Proc. IEEE 58th Vehicular Technology Conf.*, Oct. 2003, vol. 1, pp. 670–673.

[14] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed. New York: Pearson/Prentice Hall, 2004.

[15] I. C. Park and S. H. Kang, "Scheduling algorithm for partially parallel architecture of LDPC decoder by matrix permutation," in *Proc. IEEE ISCAS*, May 2005, pp. 5778–5781.

[16] T. Zhang and K. K. Parhi, "VLSI Implementation-oriented (3,k)-regular low-density parity-check codes," in *IEEE Workshop on Signal Processing Systems (SiPS)*, Sep. 2001, pp. 25–36.

[17] Y. Chen and K. K. Parhi, "Overlapped message passing for quasi-cyclic low density parity check codes," *IEEE Trans. Circuits Syst.*, vol. 51, pp. 1106–1113, Jun. 2004.

[18] S. Kim and K. K. Parhi, "Overlapped decoding for a class of quasi-cyclic codes," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, Oct. 2004, pp. 113–117.

[19] J. Li, X. H. You, and J. Li, "Early stopping for LDPC decoding: Convergence of Mean Magnitude (CMM)," *IEEE Commun. Lett.*, vol. 10, no. 9, pp. 667–669, Sep. 2006.

[20] F. Kienle and N. Wehn, "Low complexity stopping criterion for LDPC code decoders," in *Proc. 2005 IEEE Vehicular Technology Conf.*, May 2005, vol. 1, pp. 606–609.

[21] R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1117–1120, Oct. 2002.

[22] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," *IEEE J. Solid-State Circuits*, vol. 37, pp. 404–412, Mar. 2002.

[23] A. Darabiha, A. C. Carusone, and F. R. Kschischang, "Multi-Gbit/sec low density parity check decoders with reduced interconnect complexity," in *Proc. IEEE ISCAS*, May 2005, vol. 5, pp. 5194–5197.

[24] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, pp. 684–698, Mar. 2006.

[25] S. H. Kang and I. C. Park, "Loosely coupled memory-based decoding architecture for low density parity check codes," *IEEE Trans. Circuits Syst. I*, vol. 53, no. 5, pp. 1045–1056, May 2006.

[26] X. Y. Shih, C. Z. Zhan, C. H. Lin, and A. Y. Wu, "A 19-mode 8.29 mm^2 52-mW LDPC decoder chip for IEEE 802.16e system," in *Proc. Int. Symp. VLSI Circuits and VLSI Technology (SOVC-2007)*, Kyoto, Japan, Jun. 2007, pp. 16–17.

Xin-Yu Shih was born in ChangHua, Taiwan, in 1982. He received the B.S. degree from National Tsing Hua University in electrical engineering in 2004 and the M.S. degree from National Taiwan University in electronics engineering in 2006. He is now working toward the Ph. D degree in National Taiwan University.

His research interests include VLSI hardware design, digital signal processing, channel coding, and communication ICs. He is currently working on Low-density Parity-check decoder designs for advanced wire-line and wireless communication systems.

Cheng-Zhou Zhan received the B.S. degree in electronic engineering from National Taiwan University, Taipei, Taiwan, in 2005. He received the M.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan in 2007. He is working toward the Ph.D. degree in electronic engineering at National Taiwan University, Taipei, Taiwan.

His research interests are designs of very large-scale integration architectures and circuits for digital signal processing and communication systems.

Cheng-Hung Lin received the B.S. degree in electronic engineering from Fu Jen Catholic University, Taipei, Taiwan, in 2002. He received the M.S. degree in electrical engineering from National Central University, Taoyuan, Taiwan in 2004. He is working toward the Ph.D. degree in electronic engineering at National Taiwan University, Taipei, Taiwan.

His research interests include the design of very large-scale integration architectures and circuits for digital signal processing and communication systems. He is currently working on the hardware design for coding systems.

An-Yeu (Andy) Wu (S'91-M'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1987, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, in 1992 and 1995, respectively, all in electrical engineering.

From August 1995 to July 1996, he was a Member of the Technical Staff at AT&T Bell Laboratories, Murray Hill, NJ, working on high-speed transmission IC designs. From 1996 to July 2000, he was with the Electrical Engineering Department of National Central University, Taiwan. In August 2000, he joined the faculty of the Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, where he is currently a Professor. His research interests include low-power/high-performance VLSI architectures for DSP and communication applications, adaptive/multirate signal processing, reconfigurable broadband access systems and architectures, and SoC platform for software/hardware co-design.

Dr. Wu was a four-time recipient of the A-Class Research Award from the National Science Council between 1997–2000. In 2004, he received the Distinguished Young Engineer Award from The Chinese Institute of Engineers (CIE), Taiwan. In 2005, he received two research awards, the Ta-you Award (Young Scholar Award) and the President Fu Si-nien Award, from the National Science Council and National Taiwan University, respectively, for his research works in VLSI system designs.